

目录

1 引言	3
1.1 阅读对象	3
1.2 U8 二次开发概述	3
1.3 U8 使用以及二次开发现存问题	3
1.3.1 U8 使用问题	3
1.3.2 二次开发现存问题	3
1.4 编写目的	4
2 控件使用	4
2.1 开发环境	4
2.2 单据控件	4
2.2.1 如何使单据自定义项根据用户的需求弹出参照	4
2.2.2 库存单据 USER-PCO 控件的使用	14
2.2.3 单据标准（排序、合并、批改等功能）	20
2.2.4 单据参照的调用	24
2.2.5 单据打印标准化	31
2.3 单据列表控件	32
2.3.1 单据列表输出功能	32

2.3.2 单据列表的分页功能 32

2.3.3 单据列表的定位功能（列表数据定位） 37

1 引言

1.1 阅读对象

本文档针对参与用友 U8-ERP 870 及以上版本平台产品插件二次开发的技术人员，读者必须具备一定的软件开发基础，并对 U870 及以上版本插件开发技术有一定的了解。

1.2 U8 二次开发概述

目前，各个行业之间，如服装、化工、食品、机械等各个行业之间，差异越来越大，行业的细分化越来越大，各个行业都存在其比较特殊的行业特性，ERP 产品为了把开发的数量减少到最低，缩短实施周期，减少项目投入，减少实施的成本，ERP 产品必须走行业化道路才行。因此对于标准产品用友 ERP-U8，要适应行业的发展也必须走行业化的道路。

1.3 U8 使用以及二次开发现存问题

1.3.1 U8 使用问题

不具备行业化特性，针对具体的行业，尤其是服装、化工、食品等个性化鲜明的行业，U8 产品可以解决企业信息化的大部分需求，但要体现各行业需求尚存在很大的不足，需要大量二次开发才可满足，这点使 ERP-U8 很难做到普及。

1.3.2 二次开发现存问题

二次开发人员在参与 U8 的二次开发时主要存在以下问题：

1. 技术多样：就目前我了解的，U8 的二次开发采用了多种工具，多种开种语言，例如 VB、Dephil
2. 集成性差，没有 U8 原有的产品不能很好的集成，主要体现在与标准产品的界面风格，基本功能不一致
3. 获得帮助少，二次开发人员进行 U8 的二次开发时，往往需要自己研究 U8 标准产品的用法，遇到一些问题也无法得到及时的解答，只能在摸索中前进，这样导致开发的效率时常比较低下
4. 对于 U8 标准控件的使用不全面，相对局限性，二次开发人员在 U8 标准控件时，通常是只知道其一，不知道其二，许多标准控件中的功能没能及时应用上，这与 U8 标准控件没有形成一个全面的帮助有关。

1.4 编写目的

本手册主要是描述我在进行 U8 二次开发的过程中一些经验与体会，包括对 U8 标准控

件的使用以及一些问题的排错，在后续的开发中会逐渐完善该文档。

2 控件使用

2.1 开发环境

开发详述：

开发工具：VB+SQL SERVER2000

操作系统：windows2000;windows xp pro;windows 2003 (建议使用 windows2003)

U8 版本：U870+SP1

2.2 单据控件

2.2.1 如何使单据自定义项根据用户的需求弹出参照

用户在使用U8的过程中可能需要根据自己业务需求启用U8的单据的自定义项为自己的单据添加一些业务参照，这就需要开发人员在开发过程中注意把调用系统自定义项的代码添加进去。

2.2.1.1 组件引用

部件引用：

注意：单据控件有两种，一种是 U860 中使用的，它们名字的区别是 870 的是

UAPVoucherControl85，而 860 的是 VoucherControl85，两个控件切忌一起引入，不然会有冲

突，程序无法识别。

DLL 引用：

图示中红色标注的是较为重要的一些 DLL。

2.2.1.2 参考代码段：

要使新做的单据支持单据的自定义项需要在代码中添加以下代码：

第一步：'基础档案、单据参照统一过程

```
Public Sub BaseBillRef(moRef As Object, vis As clsItemState, sRet As Variant)
```

```
Dim Rst As ADODB.Recordset
```

```
Dim strGrid As String, _
```

```
strFld As String, _
```

```
strChn As String, _
```

```
strWid As String, _
```

```
strTab As String
```

```
If vis.nDataSource = 1 Then '参照基础档案
```

```
    ( sDataRule=department , cdepcode )
```

```
If moRef.EnumRefInit(g_oLogin, enuTreeViewAndGrid, False,
```

```
Left$(vis.sDataRule, InStr(1, vis.sDataRule, ",") - 1)) = False Then Exit Sub
```

```
strFld = Mid(vis.sDataRule, InStr(1, vis.sDataRule, ",") + 1)
```

```
ElseIf vis.nDataSource = 2 Then '参照单据档案
```

```
    ( sDataRule=17 , csocode ) 17 为 cardnumber
```

```
strTab = GetBillRefString(Left$(vis.sDataRule, InStr(1, vis.sDataRule, ",") - 1),
```

```
strGrid, strChn, strWid, False)
```

```
If strTab = "" Then Exit Sub
```

'返回值列没有显示，则补充。

```
If InStr(1, LCase(strGrid), LCase(Mid(vis.sDataRule, InStr(1, vis.sDataRule, ",")  
+  
1))) <= 0 Then
```

```
1))) <= 0 Then
```

```
strGrid = strGrid & "," & Mid(vis.sDataRule, InStr(1, vis.sDataRule, ",") + 1)
```

```
strChn = strChn & ",隐藏列"
```

```
strWid = strWid & ",0"
```

```
End If
```

```
strGrid = strGrid & " from " & strTab
```

```
If moRef.StrRefInit_SetColWidth(g_oLogin, False, "", strGrid, strChn, strWid)  
=  
False Then Exit Sub
```

```
False Then Exit Sub
```

```
strFld = Mid(vis.sDataRule, InStr(1, vis.sDataRule, ",") + 1)
```

```
End If
```

'统一赋值

```
moRef.Show: Set Rst = moRef.recmx
```

```
If Rst Is Nothing Then Exit Sub
```

```
sRet = Rst(strFld)
```

```
End Sub
```

第二步：以下代码是上述代码中 GetBillRefString 函数：

'获取单据参照参数 GetBillRefString

```
Public Function GetBillRefString(ByVal strCardNum As String, _  
strSelFldLst As String, _  
strChnFldLst As String, _  
strColWidLst As String, _  
ByVal bOnlyTab As Boolean) As String  
  
Dim strSql As String  
Dim strTab As String  
Dim Rst As ADODB.Recordset  
  
GetBillRefString = ""  
  
strSql = "select BTTblName from vouchers where cardnumber='" &  
strCardNum & """  
  
Set Rst = gConn.Execute(strSql)  
  
If Rst.EOF Or Rst.BOF Then Exit Function  
  
strTab = Rst(0)  
  
If bOnlyTab Then  
GetBillRefString = strTab  
  
Exit Function  
  
End If
```

```
strSelFldLst = "Select "
```

```
strSql = "select fieldname,(case when cardformula1=" then (case when  
cardformula2="
```

```
then carditemname else cardformula2 end) else cardformula1 end) as  
chnname from voucheritems
```

```
where cardnum='" & strCardNum & "' and cardsection='T' and Showit=1  
and TableName='" &
```

```
strTab & ""
```

```
Set Rst = gConn.Execute(strSql)
```

```
If Rst.EOF Or Rst.BOF Then Exit Function
```

```
Do While Not Rst.EOF
```

```
strSelFldLst = strSelFldLst & Rst!FieldName & ","
```

```
strChnFldLst = strChnFldLst & Trim(Rst!chnname) & ","
```

```
strColWidLst = strColWidLst & "1500,"
```

```
Rst.MoveNext
```

```
Loop
```

```
strSelFldLst = Left(strSelFldLst, Len(strSelFldLst) - 1)
```

```
strChnFldLst = Left(strChnFldLst, Len(strChnFldLst) - 1)
```

```
strColWidLst = Left(strColWidLst, Len(strColWidLst) - 1)
```

```
GetBillRefString = strTab
```


End Function

第三步：当添加一个单据的参照之后，要添加基础档案的检测：

'基础档案、单据检查统一过程

```
Public Function BaseBillChk(vis As clsItemState, RetValue As Variant,
bChanged As
```

```
UAPVoucherControl85.CheckRet) As Boolean
```

```
Dim strSql As String, _
```

```
strTab As String
```

```
Dim Rst As ADODB.Recordset
```

```
BaseBillChk = False
```

```
If Trim(RetValue) = "" Then
```

```
RetValue = Trim(RetValue)
```

```
bChanged = success
```

```
BaseBillChk = True
```

```
Exit Function
```

```
End If
```

```
If vis.nDataSource = 1 Then '参照基础档案
```

```
( sDataRule=department , cdepcode )
```

```
strSql = "select 1 from " & Left$(vis.sDataRule, InStr(1, vis.sDataRule, ",") - 1)
```

& "

where " & Mid(vis.sDataRule, InStr(1, vis.sDataRule, ",") + 1) & "=" & RetValue & ""

Set Rst = gConn.Execute(strSql)

If Rst.EOF Or Rst.BOF Then

MsgBox "输入的档案项不存在!", vbInformation

bChanged = Cancel

Exit Function

End If

ElseIf vis.nDataSource = 2 Then '参照单据档案

(sDataRule=17 , csocode) 17 为 cardnumber

strTab = GetBillRefString(Left\$(vis.sDataRule, InStr(1, vis.sDataRule, ",") - 1),
" , "" ,

"" , True)

If strTab = "" Then

MsgBox "单据不存在!", vbInformation

bChanged = Cancel

Exit Function

End If

strSql = "select 1 from " & strTab & " where " & Mid(vis.sDataRule, InStr(1,

```
vis.sDataRule, ",") + 1) & "=" & RetValue & ""
```

```
Set Rst = gConn.Execute(strSql)
```

```
If Rst.EOF Or Rst.BOF Then
```

```
MsgBox "输入的单据项不存在!", vbInformation
```

```
bChanged = Cancel
```

```
Exit Function
```

```
End If
```

```
End If
```

```
BaseBillChk = True
```

```
End Function
```

第四步：在单据的参照事件内添加以下代码（判断是何种类型）红字部分：

```
'自定义参照
```

```
'bill 单据对象
```

```
'r 0 表示表头参照，非 0 表示表体参照的当前行号。
```

```
'vis 表头或表体项目信息对象
```

```
'sRet 参照返回值
```

```
Public Sub Bus_BrowUser(Bill As Object, ByVal r As Long, vis As clsItemState,  
sRet As Variant)
```

```
Dim Rst As ADODB.Recordset
```

Dim strGrid As String

Dim strFld As String

Dim Rs As ADODB.Recordset

Dim sql As String

On Error GoTo errHandle

If r = 0 Then '表头参照

If vis.nDataSource = 0 Then '业务需要的参照

'这里面添加业务需要的用户自己编写的参照代码

Else

BaseBillRef reRef, vis, sRet '系统预设的参照 (单据表头自定义项)

End If

Else

If vis.nDataSource = 0 Then '判断 vis 的值判断是何种业务的参照

'这里面添加业务需要的用户自己编写的参照代码

Else

BaseBillRef reRef, vis, sRet '系统预设的参照

End If

End If

Exit Sub

errHandle:

MsgBox Err.Description

End Sub

第五步：添加数据检测，把相关代码加入单据的检测事件中（红色部分）：

'有效性检查

'bill 单据对象

'RetVal 当前值

'bChanged 检查结果取消或重试或成功

'r 0 表示表头检查，非 0 表示表体检查的当前行号。

'vis 表头或表体项目信息对象

Public Sub Bus_CellCheck(Bill As Object, RetValue As Variant, bChanged As Long, ByVal r As

Long, vis As clsItemState)

Dim Rst As ADODB.Recordset

Dim strSql As String

Dim sql As String

Dim Rs As ADODB.Recordset

On Error GoTo errHandle

If r = 0 Then '表头

If vis.nDataSource = 0 Then '业务需要的检查

'用户自己编写的参照的检测

Else

If BaseBillChk(vis, RetValue, bChanged) = False Then Exit Sub
'系统预设的检查

End If

Else

If vis.nDataSource = 0 Then '业务需要的检查

'用户自己编写的参照的检测

Else

If BaseBillChk(vis, RetValue, bChanged) = False Then Exit Sub
'系统预设的检查

End If

End If

ExtOK:

bChanged = CheckRet.success

Exit Sub

errHandle:

MsgBox Err.Description

End Sub

2.2.1.3 系统设置

当完成了以上工作之后，用户就可以第六步的设置完成自己定义的项目。

进入系统的节点 -基础设置——基础档案设置——其它：进入自定义项档案设置

设置自定义项：

完成了之后，到单据格式中设置单据体就行了具体操作如下：

自定义项属性设置：

注意：如果要使用在单据格式设置界面可以进行参照类型的设置，开发人员在开发单据时，

使用单据设置单据模板时要注意在自定义项目设置为 1：

2.2.2 库存单据 USER-PCO 控件的使用

2.2.2.1 U870 准备工作、组件引用

二次开发人员进行进行 USER-PCO 的使用之前，要在 VB 的工程引用中，引入该 DLL：

2.2.2.2 函数接口说明

新增单据- Insert 函数

参数：

VoucherType —— 单据类型，对应 VouchType 表中的 cVouchType 字段

Header —— 表头 DOM 对象

Body —— 表体 DOM 对象

Position —— 货位 DOM 对象

ErrMsg —— 错误信息

ConnFrom —— 连接对象

VoucherId —— 单据 ID

DOMMsg —— 超可用量提示信息 DOM 对象

Check —— 是否进行业务检查

BeforCheckStock —— 是否检查可用量

IsRedVouch —— 是否红字单据

ReMote —— 是否远程

删除单据-Delete 函数

参数：

VoucherType —— 单据类型，对应 VouchType 表中的 cVouchType 字段

VoucherId —— 单据 ID

ErrMsg —— 错误信息

ConnFrom —— 连接对象

TimeStamp —— 单据时间戳

DOMMsg —— 超可用量提示信息 DOM 对象

Check —— 是否进行业务检查

BeforCheckStock —— 是否检查可用量

修改单据-Update 函数

参数：

VoucherType —— 单据类型，对应 VouchType 表中的 cVouchType 字段

Header —— 表头 DOM 对象

Body —— 表体 DOM 对象

Position —— 货位 DOM 对象

ErrMsg —— 错误信息

ConnFrom —— 连接对象

VoucherId —— 单据 ID

DOMMsg —— 超可用量提示信息 DOM 对象

Check —— 是否进行业务检查

BeforCheckStock —— 是否检查可用量

IsRedVouch —— 是否红字单据

AddedState —— 修改状态

装载单据-Load 函数

参数：

VoucherType —— 单据类型，对应 VouchType 表中的 cVouchType 字段

Condition —— 条件串

Header —— 表头 DOM 对象

Body —— 表体 DOM 对象

Position —— 货位 DOM 对象

ErrMsg —— 错误信息

GetBlank —— 是否获取空白单据

BodyOrderBy —— 表体排序方式字段

审核单据-Audit 函数

参数：

VoucherType —— 单据类型，对应 VouchType 表中的 cVouchType 字段

VoucherId —— 单据 ID

ErrMsg —— 错误信息

ConnFrom —— 连接对象

TimeStamp —— 单据时间戳

DOMMsg —— 超可用量提示信息 DOM 对象

Check —— 是否进行业务检查

BeforCheckStock —— 是否检查可用量

弃审单据-CancelAudit 函数

参数：

VoucherType —— 单据类型，对应 VouchType 表中的 cVouchType 字段

VoucherId —— 单据 ID

ErrMsg —— 错误信息

ConnFrom —— 连接对象

TimeStamp —— 单据时间戳

DOMMsg —— 超可用量提示信息 DOM 对象

Check —— 是否进行业务检查

BeforCheckStock —— 是否检查可用量

2.2.2.3 参考代码 (Insert 方法)

1. 使用该函数，二次开发人员还得对存货的可用量进行判断以及获取系统的设置，看是否

设置了可超可用量发货，然后设置相应的是否检查业务与是否检查可用量，参考代码如下：

```
*****
```

```
'判断是否进行可用量的检查 *'
```

```
'by 王冬冬 2007-11-29 *'
```

```
*****
```

```
Public Function CheckKQty() As Boolean
```

```
Dim bNoBatch As Boolean
```

```
Dim bYesBatch As Boolean
```

```
Dim strSql As String
```

```
Dim rs As New ADODB.Recordset
```

'查询表体的存货是否存在非批次管理的

```
strSql = "select * from HY_FZ_SendMaterialOrder_details inner join  
inventory on
```

```
HY_FZ_SendMaterialOrder_details.cInvCode=inventory.cInvCode where  
bInvBatch=0 and id=""
```

```
& Voucher.headerText("ID") & ""
```

```
rs.Open strSql, g_Conn, 1, 1
```

```
If rs.RecordCount > 0 Then
```

```
bNoBatch = True '存在非批次管理标志
```

```
Else
```

```
bNoBatch = False '不存在非批次管理标志
```

```
End If
```

```
rs.Close
```

'查询表体的存货是否存在批次管理

```
strSql = "select * from HY_FZ_SendMaterialOrder_details inner join  
inventory on
```

```
HY_FZ_SendMaterialOrder_details.cInvCode=inventory.cInvCode where  
bInvBatch=1 and id=""
```

```
& Voucher.headerText("ID") & ""
```

```
rs.Open strSql, g_Conn, 1, 1
```

```
If rs.RecordCount > 0 Then
```

```
bYesBatch = True '存在批次管理标志
```

Else

bYesBatch = False '不存在批次管理标志

End If

rs.Close

'当表体存货中两种类型存货都存在，则必须检查两种设置

If bNoBatch = True And bYesBatch = True Then

strSql = "select cValue from AccInformation where cSysID='ST' and
cName='bAllowZero' or cName='bBatchAllowZero'"

rs.Open strSql, g_Conn, 1, 1

If LCase(rs.Fields("cValue")) = "true" Then

rs.MoveNext

If LCase(rs.Fields("cValue")) = "true" Then

CheckKQty = False

Else

CheckKQty = True

End If

Else

CheckKQty = True

End If

rs.Close

'当表体只存在批次管理存货，则检查批次管理的存货是否允许超可用量出货

ElseIf bNoBatch = False And bYesBatch = True Then

strSql = "select cValue from AccInformation where cSysID='ST' and
cName='bBatchAllowZero'"

rs.Open strSql, g_Conn, 1, 1

If LCase(rs.Fields("cValue")) = "true" Then

CheckKQty = False

Else

CheckKQty = True

End If

rs.Close

'当表体只存在非批次管理存货，则检查非批次管理的存货是否允许超可用量出货

ElseIf bNoBatch = True And bYesBatch = False Then

strSql = "select cValue from AccInformation where cSysID='ST' and
cName='bAllowZero'"

rs.Open strSql, g_Conn, 1, 1

If LCase(rs.Fields("cValue")) = "true" Then

CheckKQty = False

Else

```
CheckKQty = True
```

```
End If
```

```
rs.Close
```

```
End If
```

```
End Function
```

该函数返回是否检查业务与可用量的标志值

2. 然后就开始对 USER-PCO 进行使用配置。（参考代码）

```
Public Function CreateOut(HTable As String, BTable As String) As Boolean
```

```
Dim sql As String, sSql As String
```

```
Dim rs As ADODB.Recordset
```

```
Dim sID As String
```

```
Dim sSource As String
```

```
Dim odomhead As New DOMDocument
```

```
Dim sDomHead As New DOMDocument
```

```
Dim odombody As New DOMDocument
```

```
Dim sDombody As New DOMDocument
```

```
Dim node As IXMLDOMElement
```

```
Dim aNode As IXMLDOMElement
```

```
Dim oDomPosition As DOMDocument
```

Dim eleHead As IXMLDOMElement

Dim seleHead As IXMLDOMElement

Dim eleBody As IXMLDOMElement

Dim seleBody As IXMLDOMElement

Dim sNode As IXMLDOMNode

Dim oNode As IXMLDOMNode

Dim cinvcode As String

Dim cinname As String

Dim color As String

Dim size As String

Dim pmMatch As String

Dim errMsg As String

Dim Flag As Boolean

Dim bCheck As Boolean

'检查是否进行可用量检查

bCheck = CheckKQty

'构建表头 XML,对应于主表 RdRecord

sql = "Select * From RecordOutQ where 1=2"

Set rs = g_Conn.Execute(sql)


```
rs.Save odomhead, adPersistXML
```

```
rs.Close
```

```
sSql = "Select * From " & HTable 'HTable 为单据数据主表
```

```
Set rs = g_Conn.Execute(sSql)
```

```
rs.Save sDomHead, adPersistXML
```

```
rs.Close
```

```
Set seleHead = sDomHead.selectSingleNode("//z:row")
```

```
Set eleHead = odomhead.selectSingleNode("//rs:data")
```

```
eleHead.appendChild seleHead
```

```
' Set eleHead = oDomhead.selectSingleNode("//z:row")
```

```
'eleHead.setAttribute "cCode", GetMaxValue(HTable, "cCode")
```

```
'构建表体 XML,对应于表 RdRecords
```

```
sSql = "Select * From RecordOutsQ where 1=2"
```

```
Set rs = g_Conn.Execute(sSql)
```

```
rs.Save odombody, adPersistXML
```

```
rs.Close
```

```
sSql = "Select * From " & BTable 'BTable 为单据数据子表
```

```
Set rs = g_Conn.Execute(sSql)
```

```
rs.Save sDombody, adPersistXML
```

```
rs.Close
```

```
Set sNode = sDombody.selectSingleNode("//rs:data")
```

```
Set oNode = odombody.selectSingleNode("//rs:data")
```

```
oNode[xss_clean].appendChild sNode
```

```
Set oNode = odombody.selectSingleNode("//rs:data")
```

```
oNode[xss_clean].removeChild oNode
```

```
Dim objPco As New USERPCO.VoucherCO
```

```
Dim a As Object
```

```
If g_oLogin.LoginEx("ST", a) = False Then
```

```
MsgBox "无法登陆库存子系统，无法删除单据！"
```

```
Exit Function
```

```
End If
```

```
Call objPco.IniLogin(g_oLogin, sSource)
```

```
Dim ErDom As DOMDocument
```

```
Flag = objPco.Insert("11", odomhead, odombody, oDomPosition, sSource, ,  
sID, ErDom,
```

```
bCheck, bCheck)
```

```
'下面对超可用量进行判断，解析
```

```
If Flag = False And sSource = "" Then
```

```
Set node = ErDom.selectSingleNode("//zeroout")
```

```
For Each aNode In node.selectNodes("//z:row")
    cinvcode = CStr(aNode.getAttributeNode("cinvcode").Value)
    cinvname = CStr(aNode.getAttributeNode("cinvname").Value)
    color = CStr(aNode.getAttributeNode("cfree1").Value)
    size = CStr(aNode.getAttributeNode("cfree2").Value)
    pmMatch = CStr(aNode.getAttributeNode("cfree3").Value)
    errMsg = errMsg & "存货编码：" & cinvcode & " 存货名称：" & cinvname &
    "
    颜色：" & color & " 尺码：" & size & " 配码：" & pmMatch & vbCrLf
Next
MsgBox errMsg & vbCrLf & "以上信息的产品超可用量出库，生单失败",
vbInformation, "生单失败!"
CreateOut = False
Exit Function
ElseIf sSource <> "" Then
MsgBox sSource, vbInformation, "生单失败"
CreateOut = False
Exit Function
Else
'返写生成的单据号
```

```
g_Conn.Execute " update HY_FZ_SendMaterialOrder_main set  
outcode=(select ccode  
  
from rdrecord where id=" & sID & "),outid = " & sID & " ,vouchtypename =  
'材料出库单'  
  
where id=" & Voucher.headerText("id")  
  
MsgBox "生单成功！", vbInformation, "提示"  
  
CreateOut = True  
  
ExecRefresh  
  
End If  
  
End Function
```

2.2.3 单据标准（排序、合并、批改等功能）

（注：单据标准说明由徐燕编写以及提供资料，特此致谢）

U8 标准产品单据控件集合了很多好的应用，给用户的操作带来很大的便利性，这里列举几种常见的应用：合并显示、表体定位、表体排序、批量修改、打印、输出，帮助

2.2.3.1 合并显示

参数说明：vch 单前操作单据对象

变量说明：g_oLogin 登陆对象

```
Public Sub ExecShowAgg(vch as ctlVoucher) '合并显示
```

```
On Error GoTo errHandle
```

```
vch.ProtectUnload2
```

```
Dim m_oDataSource As Object
```

```
Set m_oDataSource = CreateObject("IDataSource.DefaultDataSource")
```

```
If m_oDataSource Is Nothing Then
```

```
MsgBox "无法创建 m_oDataSource 对象!", vbExclamation
```

```
Exit Sub
```

```
End If
```

```
Set m_oDataSource.SetLogin = g_oLogin
```

```
Set vch.SetDataSource = m_oDataSource
```

```
vch.SHowAggregateSetupDlg
```

```
Exit Sub
```

```
errHandle:
```

```
DefineMsgBox Err.Description + " , 表体行处理失败！"
```

```
End Sub
```

注：合并显示在 U872 以上版本还添加以 CHECKBOX 方式勾选是否合并显示具体实现如下：

在单据加载模板之前调用：

```
Voucher.ShowSummaryView=True
```

2.2.3.2 表体定位

调用单据自身函数：Voucher.ShowFindDlg

2.2.3.3 表体排序

(1) 在 form_load 中设置 Voucher.ShowSorter = True

2.2.3.4 批量修改

'参数说明：vch 单前操作单据对象

'变量说明：g_oLogin 登陆对象

```
Public Sub ExecBatchModify(vch As ctlVoucher) '批量修改
```

```
On Error GoTo errHandle
```

```
Dim m_oDataSource As Object
```

```
Set m_oDataSource = CreateObject("IDataSource.DefaultDataSource")
```

```
If m_oDataSource Is Nothing Then
```

```
MsgBox "无法创建 m_oDataSource 对象!", vbExclamation
```

```
Exit Sub
```

```
End If
```

```
Set m_oDataSource.SetLogin = g_oLogin
```

```
Set vch.SetDataSource = m_oDataSource
```

```
vch.ShowBatchModify
```

```
Exit Sub
```

errHandle:

DefineMsgBox Err.Description + " , 表体行处理失败 ! "

End Sub

2.2.3.5 打印

'参数说明 :

'oConnection (ADODB.Connection) 数据库连接对象

'oVoucher (ctlVoucher) 单据对象

'sBillNumber (String) 单据号

'sTemplateID (String) 模板号

'bPreview [Boolean, False] 标志是否显示预览界面

Public Sub VoucherPrint(_

ByRef oConnection As ADODB.Connection, _

ByRef oVoucher As ctlVoucher, _

ByVal sBillNumber As String, _

ByVal sTemplateID As String, _

Optional ByVal bPreview As Boolean = False)

Dim oField As ADODB.Recordset ' 固定文本数据

```
Dim oTemplate As ADODB.Recordset ' 单据模板数据
```

```
Dim oVoucherTemplate As UFVoucherServer85.clsVoucherTemplate
```

```
Dim sError As String
```

```
Dim oDomhead As DOMDocument
```

```
Dim oDombody As DOMDocument
```

```
Set oVoucherTemplate _
```

```
= CreateObject("UFVoucherServer85.clsVoucherTemplate")
```

```
If oVoucherTemplate Is Nothing Then
```

```
MsgBox "创建模板对象失败。", vbCritical
```

```
GoTo Exit_Label
```

```
End If
```

```
Set oTemplate = oVoucherTemplate.GetTemplateData2( _
```

```
Conn:=oConnection, _
```

```
sBillName:=sBillNumber, _
```

```
vTemplateID:=sTemplateID)
```

```
Set oField = oVoucherTemplate.GetFixedData( _
```

```
Conn:=oConnection.ConnectionString, _
```


vVtid:=sTemplateID)

Call oVoucher.PrintVoucher(_

rsTemplate:=oTemplate, _

rsField:=oField, _

bShowPrintViewDlg:=bPreview)

Exit_Label:

On Error GoTo 0

Set oDomhead = Nothing

Set oDombody = Nothing

Set oVoucherTemplate = Nothing

If Not oField Is Nothing Then

If oField.state = adStateOpen Then _

Call oField.Close

End If

Set oField = Nothing

If Not oTemplate Is Nothing Then

If oTemplate.state = adStateOpen Then _

Call oTemplate.Close

End If

Set oTemplate = Nothing

End Sub

2.2.3.6 输出

'导出单据单据数据到指定的文件

'参数说明：

'oConnection (ADODB.Connection) 数据库连接对象

'oVoucher (ctlVoucher) 单据对象

'sBillNumber (String) 单据号

'sTemplateID (String) 模板号

Public Sub ExportVoucherData2File(_

ByRef oConnection As ADODB.Connection, _

ByRef oVoucher As ctlVoucher, _

ByVal sBillNumber As String, _

ByVal sTemplateID As String)

Dim oField As ADODB.Recordset ' 固定文本数据

Dim oTemplate As ADODB.Recordset ' 单据模板数据

Dim oVoucherTemplate As Object

'On Error GoTo Err_Handler

Set oVoucherTemplate _

= CreateObject("UFVoucherServer85.clsVoucherTemplate")

If oVoucherTemplate Is Nothing Then

MsgBox "创建模板对象失败。", vbCritical

GoTo Exit_Label

End If

Set oTemplate = oVoucherTemplate.GetTemplateData2(_

Conn:=oConnection, _

sBillName:=sBillNumber, _

vTemplateID:=sTemplateID)

Set oField = oVoucherTemplate.GetFixedData(_

Conn:=oConnection.ConnectionString, _

vVtid:=sTemplateID)

Call oVoucher.ExportToFile(_

rsTemplate:=oTemplate, _

rsField:=oField)

Exit_Label:

On Error GoTo 0

Set oVoucherTemplate = Nothing

If Not oField Is Nothing Then

If oField.state = adStateOpen Then _

Call oField.Close

End If

Set oField = Nothing

If Not oTemplate Is Nothing Then

If oTemplate.state = adStateOpen Then _

Call oTemplate.Close

End If

Set oTemplate = Nothing

End Sub

2.2.4 单据参照的调用

2.2.4.1 单据定义参照调用方法

2.2.4.1.1 组件引用

参照服务：

2.2.4.1.2 参考代码

```
Private hwRef As New UFRReferC.UFRReferClient
```

```
'声明一个参照服务客户端的对象
```

```
'自定义参照
```

```
'Bill 单据对象
```

```
'vis 表头或表体项目信息对象
```

```
'sRet 参照返回值
```

```
Public Sub Bus_BrowUser(Bill As Object, ByVal r As Long, ByVal C As Long,  
sRet As
```

```
Variant)
```

```
Dim rst As ADODB.Recordset
```

```
Dim strGrid As String
```

```
Dim strFld As String
```

```
Dim strRet As String
```

```
Dim sField As String
```

```
InvCodePos = Bill.GetColIndex(cInvCode)
```

```
InvNamePos = Bill.GetColIndex(cInvName)
```

```
colorPos = Bill.GetColIndex(color)
```

```
sizePos = Bill.GetColIndex(size)
```

```
BCodePos = Bill.GetColIndex(BCode)
```

```
PartIdPos = Bill.GetColIndex(PartID)
```

```
cInvDefinePos = Bill.GetColIndex(cInvDefinePos)
```

```
On Error GoTo errHandle
```

```
sField = Bill.GetColName(C)
```

```
Select Case sField
```

```
Case cInvCode
```

```
If hwRef.EnumRefInit(g_oLogin, enuTreeViewAndGrid, False,  
DataType.enuStockInven) =
```

```
False Then Exit Sub '调用系统封装好的参照 ( DataType )
```

```
strFld = "cInvCode"
```

```
Case color, size
```

```
If Bill.TextMatrix(r, InvCodePos) = "" Then
```

```
MsgBox "请先选择产品!", , "提示"
```

```
Exit Sub
```

```
Else
```

```
strGrid = "select partid,InvCode,cInvName,Free1,Free2 from
```

```
V_DetailInventory where invcode ='" & Bill.TextMatrix(r, InvCodePos) & """
```

```
If hwRef.StrRefInit_SetColWidth(g_oLogin, False, "", strGrid, "序号,产品
```

```
编码,产品名称,颜色,尺码", "1500,1500") = False Then Exit Sub '采用 SQL  
语句自定义参照
```

```
strFld = IIf(sField = color, "Free1", "Free2")
```

```
End If
```

```
End Select
```

```
hwRef.Show: Set rst = hwRef.recmx '显示参照
```

```
If rst Is Nothing Then Exit Sub
```

```
sRet = rst(strFld)
```

```
'todo 设置联动
```

```
Select Case sField
```

```
Case cInvCode
```

```
Bill.TextMatrix(r, C) = rst!cInvCode & ""
```

```
Bill.TextMatrix(r, InvNamePos) = rst!cInvName & ""
```

```
' Bill.TextMatrix(R, cInvDefinePos) = rst!cInvDefine10 & ""
```

```
Case color, size
```

```
Bill.TextMatrix(r, colorPos) = rst!Free1 & ""
```

```
Bill.TextMatrix(r, sizePos) = rst!Free2 & ""
```

```
End Select
```

```
Exit Sub
```

errHandle:

DefineMsgBox err.Description

End Sub

2.2.4.2 调用 UAP 设计好的系统参照 (一)

使用这个 UAP 设计好的系统参照同样需要引入参照服务的同样的 DLL

参考代码 :

```
Private Sub Voucher_headBrowUser(ByVal Index As Variant, sRet As Variant, referPara As
```

```
UAPVoucherControl85.ReferParameter)
```

```
Dim vis As UAPVoucherControl85.clsItemState
```

```
Set vis = Voucher.ItemState(Index, 0)
```

```
Dim strRefTable As String
```

```
Dim sqlstr As String
```

```
Dim color As String
```

```
Dim rs As New ADODB.Recordset
```

```
Dim rst As ADODB.Recordset
```

```
Dim strGrid As String
```

```
Dim strFld As String
```

```
Dim sHeadItemName As String
```

```
Dim RefRs As ADODB.Recordset
```



```
Dim i, j As Long
```

```
Dim sMetaXML As String
```

```
sMetaXML = "<Ref><RefSet bAuth='0' /></Ref>"
```

```
On Error GoTo errHandle
```

```
moRef.SetLogin g_oLogin
```

```
sHeadItemName = LCase(Voucher.ItemState(Index, siheader).sFieldName)
```

```
'部门参照
```

```
If sHeadItemName = "deptcode" Then
```

```
strFilterSQL = ""
```

```
referPara.ReferMetaXML = sMetaXML
```

```
referPara.ID = "Department_AA"
```

```
referPara.sSql = strFilterSQL
```

```
End If
```

```
'业务员参照
```

```
If sHeadItemName = "busyman" Then
```

```
strFilterSQL = ""
```

```
referPara.ReferMetaXML = sMetaXML
```

```
referPara.ID = "Person_AA"
```

```
referPara.sSql = strFilterSQL
```

End If

'仓库参照

If sHeadItemName = "inwarehouse" Or sHeadItemName = "outwarehouse"
Then

strFilterSQL = ""

referPara.ReferMetaXML = sMetaXML

referPara.ID = "Warehouse_AA"

referPara.sSql = strFilterSQL

End If

'出库类型

If sHeadItemName = "crdcode" Then

strFilterSQL = "bRdFlag=0 and bRdEnd=1"

referPara.ReferMetaXML = sMetaXML

referPara.ID = "Rd_Style_ST"

referPara.sSql = strFilterSQL

End If

Exit Sub

errHandle:

MsgBox Err.Description, , "提示"

End Sub

注意：在使用这种方式需要在使用该功能前设置数据源（建议放在 FORM 的 LOAD 事件内），

参考代码：

'加载计划单数据来源

Set oDataSource = CreateObject("IDataSource.DefaultDataSource")

If oDataSource Is Nothing Then

MsgBox "创建单据数据源对象 Fail!", vbExclamation

End If

Set oDataSource.SetLogin = g_oLogin

ctlVoucher2.LoginObj = g_oLogin

ctlVoucher2.InitDataSource

2.2.4.3 调用 UAP 设计好的系统参照（二）标准产品使用

组件调用：参照服务的服务器端

参考代码：（该代码支持多选）

'采用新方式调用系统参照

'编写者：王冬冬

Private Sub Voucher_bodyBrowseUser(ByVal Row As Long, ByVal Col As Long, sRet As Variant,

referPara As UAPVoucherControl85.ReferParameter)

Dim vis As UAPVoucherControl85.clsItemState

'使用新的参照服务

Dim objRefer As New U8RefService.IService '定义参照服务服务端

Set vis = Voucher.ItemState(Col, 1) '记住此处，是关键

Dim sqlstr As String

Dim rstClass As New ADODB.Recordset

Dim rstGrid As New ADODB.Recordset

Dim ErrMsg As String

Dim sBodyItemName As String

Dim sMetaXML As String

'这句重要，将参照事件设置为不启动状态

referPara.Cancel = True

'设置参照是否多选

sMetaXML = "<Ref><RefSet bAuth='0' bMultiSel= '1' /></Ref> "

moRef.SetLogin g_oLogin

sBodyItemName = LCase(Voucher.ItemState(Col, sibody).sFieldName)

If sBodyItemName = "cinvcode" Then

objRefer.RefID = "Inventory_AA"

objRefer.MetaXML = sMetaXML

'下面代码是调用参照服务（U8 标准产品）

```
If objRefer.ShowRef(g_oLogin, rstClass, rstGrid, ErrMsg) = False Then Exit Sub
```

```
If rstGrid Is Nothing Then Exit Sub
```

```
Dim rr As Long, AddNewLn As Boolean
```

```
Dim temp As String
```

```
'判断弹出的参照是否有选择或者选择取消按钮
```

```
If rstGrid.state = 0 Then Exit Sub
```

```
sRet = rstGrid("cinvcod")
```

```
rr = 0
```

```
'下列函数是对多选的数据进行处理
```

```
AddNewLn = (Voucher.Row = Voucher.bodyRows)
```

```
If rstGrid.RecordCount > 1 And Voucher.Row < Voucher.bodyRows Then
```

```
AddNewLn = (MsgBox("是否把选定存货追加到最后吗?" & Chr(13) & _
```

```
"选是将修改当前行,并把其余记录追加到最后!" & Chr(13) & _
```

```
"选否将从当前行开始覆盖!", vbYesNo + vbQuestion, "请选择编辑方
```

```
式") = vbYes)
```

```
End If
```

```
While rstGrid.EOF = False
```

```
If rr = 0 Then
```

```
Voucher.bodyText(Row + rr, "cinvcod") = rstGrid("cinvcod")
```

temp = rstGrid("cinvcode")

Fun.Bus_CellCheck Voucher, rstGrid("cinvcode"), 2, Row + rr, vis

Else

If AddNewLn Then

If Voucher.BodyRowIsEmpty(Voucher.bodyRows) = False Then

Voucher.AddLine Voucher.bodyRows + 1

Voucher.bodyText(Voucher.bodyRows - 1, "cinvcode") = temp

End If

Voucher.UpdateLineData Voucher.GetLineDom(r), Voucher.bodyRows

Voucher.bodyText(Voucher.bodyRows, "cinvcode") = rstGrid("cinvcode")

temp = rstGrid("cinvcode")

Fun.Bus_CellCheck Voucher, rstGrid("cinvcode"), 2, Voucher.bodyRows, vis

Else

Voucher.bodyText(Row + rr, "cinvcode") = rstGrid("cinvcode")

temp = rstGrid("cinvcode")

Fun.Bus_CellCheck Voucher, rstGrid("cinvcode"), 2, Row + rr, vis

End If

End If

rstGrid.MoveNext

```
rr = rr + 1
```

```
Wend
```

```
Exit Sub
```

```
End If
```

```
End Sub
```

2.2.5 单据打印标准化

单据打印标准化指二次开发出来的单据拥有与 U8 标准单据相一致的打印、输出、预览。U8 单据打印包含两方面内容：

(1) 打印格式设置：

打印格式设置，需要使用单据标准事件：SaveSettingEvent

```
Private Sub Voucher_SaveSettingEvent(ByVal varDevice As Variant)
```

```
Dim TmpUFTemplate As Object
```

```
Set TmpUFTemplate =  
CreateObject("UFVoucherServer85.clsVoucherTemplate")
```

```
If TmpUFTemplate.SaveDeviceCapabilities(g_Conn.ConnectionString,
```

```
m_strVT_PRN_ID, varDevice) <> 0 Then
```

```
MsgBox "U8.SA.xsglsql.01.frbillvouch.00361", vbInformation,
```

```
MapResidToString("U8.FZ.HY_FZ_ProSendMaterial.frm040") 'zh-  
CN：打印设置
```

```
保存失败
```

```
End If
```

End Sub

(2) 打印次数、打印密码控制：

打印次数、打印密码控制指在 U8 系统——基础设置——单据设置——单据打印控制中可设置相应单据的打印次数与打印密码。

具体实现如下：

将单据注册表内 VOUCHERS_BASE 的 IsPrintLimited 字段设置为 1

例如：

```
update VOUCHERS_BASE set IsPrintLimited =1 where  
cardnumber='FZ0205001'
```

2.3 单据列表控件

2.3.1 单据列表输出功能

在参与二次开发的途中，我发现二次开发的单据列表对输出功能都没有进行使用前设

置，导致在使用该单据列表的输出功能时，造成系统崩溃。

大家都清楚，使用单据列表输出功能只需调用单据列表控件的 PrintToFile 函数，但是在

使用该函数之前，要进行一个单据类表模板的设置才能正确输出：

对齐方式的设置（一般二次开发人员很少设置该项目）：

注意：二次开发人员如果要使用该控件的输出功能，请务必设置该对齐方式。如果不设置该

对齐方式会造成 U8 的异常退出。

2.3.2 单据列表的分页功能

2.3.2.1 组件引用

部件引用：

DLL 引用：

2.3.2.2 概述

单据列表的分页功能，二次开发人员是不需要自己设计的，只需要调用 U8

原标准产品的分页控件即可，效果展示：

2.3.2.3 功能实现

二次开发人员在引入单据列表控件部件时，在部件显示栏中会同时出现一个名叫—

pageDivCtl

的控件，单据列表的分页功能就是使用该控件，下面有详细的代码参考：

声明一些必要的变量（红色重要）：

```
Public objColset As U8ColumnSet.clsColSet
```

'常量

```
Private Const strVOUCH_KEY As String = "FZ0205001"
```

```
Private Const strDBVName As String = "v_HY_FZ_SendMaterialOrder_main"
```

Private objListRec As ADODB.Recordset

Dim recclass As New ADODB.Recordset

Private WithEvents m_pagediv As Pagediv '分页

Private m_coni As IPagedivConi '条件，基本上都是从 U8Colset 中进行初始化

Public vchVoucher As ctlVoucher

Dim m_bshowSumType As Boolean '是否汇总显示

'窗体加载函数：

Private Sub Form_Load()

On Error GoTo ErrorHandler

CurOpStatus = SHOW_ALL

Set UFToolbar1.Business = g_obusiness

init_UFtoolbar

'调整控件布局

Call FormLayout

'初始化表体

Call InitList

'填充数据

Set m_pagediv = New Pagediv '设置新的分页对象

Me.VouchList1.AdJustGridWidth

```
VouchList1.SumStyle = vlGridSum
```

```
' GetDatas
```

```
' LoadHelpId ("12250036")
```

```
ExecLocal
```

```
Exit Sub
```

```
ErrorHandler:
```

```
DefineMsgBox "错误号：" & Err.Number & "错误描述：" & Err.Description
```

```
End Sub
```

重要是下一函数：

```
'获得数据
```

```
Private Sub GetDatas()
```

```
Dim strwhere As String
```

```
Me.VouchList1.SetVchLstRst Nothing
```

```
Me.VouchList1.FillMode = FillAppend
```

```
If bFilter = True Then
```

```
strwhere = "ID in " & lngVoucherID
```

```
Else
```

```
strwhere = ""
```

```
End If
```

InitConi strwhere

Call PagedivCtl1.BindPagediv(m_pagediv)

Call m_pagediv.Initialize(g_Conn, m_coni)

m_pagediv.LoadData

Me.VouchList1.AdJustGridWidth

VouchList1.SumStyle = vlGridSum

Exit Sub

Exit_Label:

If Not objListRec Is Nothing Then

If objListRec.state = adStateOpen Then objListRec.Close

End If

End Sub

'分页事件，在得到页数之前

Private Sub m_pagediv_BeforeGetCount()

VouchList1.FillMode = FillOverwrite

End Sub

'分页事件，在得到页数之后

Private Sub m_pagediv_BeforeGetCount()

VouchList1.FillMode = FillOverwrite

End Sub

'分页事件，得到数据

```
Private Sub m_pagediv_GetData(ByVal vtable As  
U8VouchList.VouchListTable)
```

```
VouchList1.SetVchLstRst vtable.DataRecordset
```

```
Set recclass = vtable.DataRecordset
```

```
VouchList1.SetSumRst vtable.SumRecordset
```

```
VouchList1.RecordCount = vtable.DataCount
```

End Sub

'分页事件，得到数据之后，设置列表

```
Private Sub m_pagediv_AfterGetData(rst As ADODB.Recordset, cnt As Long)
```

```
Me.VouchList1.InitHead objColset.getColInfo()
```

```
VouchList1.SumStyle = vlGridSum
```

End Sub

'分页控件的事件，当转页数时触发

```
Private Sub PagedivCtl1_BeforeSendCommand(cmdType As  
U8VouchList.UFCommandType,
```

```
pageSize As Long, pagecurrent As Long)
```

```
Me.VouchList1.SetVchLstRst Nothing
```

```
Me.VouchList1.FillMode = FillOverwrite
```

```
pagecurrent1 = pagecurrent
```

```
End Sub
```

```
'初始化分页条件
```

```
Private Sub InitConi(strwhere As String)
```

```
Dim i As Long
```

```
Dim s As String
```

```
If m_coni Is Nothing Then
```

```
Set m_coni = New DefaultPagedivConi
```

```
End If
```

```
m_coni.From = strDBVName '相当于 From 部分
```

```
objColset.setColMode (strVOUCH_KEY), 0
```

```
If m_bshowSumType Then
```

```
m_coni.SelectConi = Replace(objColset.GetSqlSumString, "", " ")
```

```
If strwhere = "" Then strwhere = "1=1"
```

```
m_coni.Where = strwhere & " " & objColset.GetSqlGroupString
```

```
m_coni.OrderID = objColset.GetOrderString
```

```
Else
```

```
m_coni.SelectConi = objColset.GetSqlString
```

```
m_coni.OrderID = objColset.GetOrderString
```

m_coni.Where = strwhere '相当于 where 部分

End If

End Sub

以上代码均是参考性代码，主要的流程就是按照这个流程走。

2.3.3 单据列表的定位功能（列表数据定位）

该功能比较简单只要设置单据列表一个函数：

VouchList1.Locate True

VouchList1 是单据列表控件的名称。

待续。。。。。